



**DevOps Shack**

# **JENKINS CLI & PARAMETERS**

## **Introduction to Jenkins CLI**

- **Overview:**
  - Explain what Jenkins CLI is and why it's useful.
  - Describe how Jenkins CLI allows you to interact with your Jenkins instance from the command line.

## **2. Downloading Jenkins CLI**

- **Command:**

```
curl -o jenkins-cli.jar JENKINS_URL/jnlpJars/jenkins-cli.jar
```

- **Explanation:**
  - Detail how to download the jenkins-cli.jar from the Jenkins server.
  - Explain the significance of the URL provided.

## **3. Setting Up Environment Variables**

- **Commands:**

```
export JENKINS_URL=http://13.126.85.125:8080/
```

```
export JENKINS_USER=aditya
```

```
export JENKINS_PASSWORD=aditya
```

- **Explanation:**
  - Describe how to set up environment variables to simplify CLI commands.
  - Highlight the importance of keeping these values secure, especially the JENKINS\_PASSWORD.

## **4. Listing Jenkins Jobs**

- **Command:**

```
java -jar jenkins-cli.jar -auth $JENKINS_USER:$JENKINS_PASSWORD list-jobs
```

- **Explanation:**
  - Explain how this command lists all jobs available on the Jenkins server.

- Discuss what output to expect and how it can be used.

## 5. Building a Jenkins Job

- **Command:**

```
java -jar jenkins-cli.jar -auth $JENKINS_USER:$JENKINS_PASSWORD build job_name
```

- **Explanation:**

- Provide details on how to trigger a build for a specific job.
- Explain the parameters that can be passed to the job if needed.

## 6. Deleting a Jenkins Job

- **Command:**

```
java -jar jenkins-cli.jar -auth $JENKINS_USER:$JENKINS_PASSWORD delete-job my-new-job
```

- **Explanation:**

- Describe how to delete a job using Jenkins CLI.
- Discuss the implications and precautions to take before deleting a job.

## 7. Using Jenkins CLI Without Environment Variables

- **Commands:**

bash

Copy code

```
java -jar jenkins-cli.jar -s http://13.234.238.69:8080 -auth user:pass list-jobs
```

```
java -jar jenkins-cli.jar -s http://13.234.238.69:8080 -auth user:pass build job_name
```

```
java -jar jenkins-cli.jar -s http://13.234.238.69:8080 -auth user:pass delete-job my-new-job
```

- **Explanation:**

- Show how to run the same commands without setting environment variables.
- Discuss scenarios where this might be preferred or necessary.

## 8. Advanced CLI Commands

- **Additional Options:**

- Explore other Jenkins CLI commands such as get-job, create-job, disable-job, and enable-job.
- Provide examples and explanations for each.

# Jenkins pipeline parameters

## 1. Introduction to Jenkins Pipeline Parameters

- **Overview:**

- Explain what pipeline parameters are and their purpose in Jenkins.
- Describe how parameters allow you to customize builds and make pipelines more flexible and dynamic.

## 2. Types of Parameters

- **Overview of Available Parameters:**

- Briefly introduce the different types of parameters that can be used in Jenkins pipelines: String, Boolean, Choice, Password, File, Text, and Credentials.

## 3. String Parameter

- **Syntax:**

`string(name: 'STRING_PARAM', defaultValue: 'default value', description: 'Enter a string value')`

- **Explanation:**

- Describe the string parameter and its use for accepting free-text input.
- Discuss the attributes: name, defaultValue, and description.
- Provide examples of common use cases where string parameters are beneficial.

## 4. Boolean Parameter

- **Syntax:**

`booleanParam(name: 'BOOLEAN_PARAM', defaultValue: true, description: 'Enable or disable something')`

- **Explanation:**

- Explain the booleanParam for accepting true/false values.
- Discuss scenarios where boolean parameters are useful, such as enabling or disabling features or steps in a pipeline.

## 5. Choice Parameter

- **Syntax:**

`choice(name: 'CHOICE_PARAM', choices: ['Option 1', 'Option 2', 'Option 3'], description: 'Select an option')`

- **Explanation:**

- Describe the choice parameter, which provides a dropdown menu for selecting one of several options.
- Highlight the importance of defining meaningful choices and how they can control the flow of the pipeline.

## 6. Password Parameter

- **Syntax:**

```
password(name: 'PASSWORD_PARAM', defaultValue: 'password', description: 'Enter your password')
```

- **Explanation:**

- Discuss the password parameter and how it securely handles sensitive information.
- Explain how passwords are masked in logs to protect confidential data.

## 7. File Parameter

- **Syntax:**

```
file(name: 'FILE_PARAM', description: 'Upload a file')
```

- **Explanation:**

- Explain the file parameter, which allows users to upload files that can be used in the pipeline.
- Provide examples of use cases, such as uploading configuration files or scripts that are required during the build.

## 8. Text Parameter

- **Syntax:**

```
text(name: 'TEXT_PARAM', defaultValue: 'default text', description: 'Enter multi-line text')
```

- **Explanation:**

- Describe the text parameter, which is designed for multi-line text input.
- Highlight how it can be useful for passing long strings, such as JSON, XML, or large blocks of text.

## 9. Credentials Parameter

- **Syntax:**

```
credentials(name: 'CREDENTIALS_PARAM', description: 'Select credentials', credentialType: 'com.cloudbees.plugins.credentials.impl.UsernamePasswordCredentialsImpl')
```

- **Explanation:**

- Discuss the credentials parameter, which integrates with Jenkins' credentials management to securely pass sensitive information.
- Explain how different types of credentials (e.g., username/password, SSH keys) can be used and retrieved within the pipeline.

## 10. Using Parameters in Stages

- **Example:**

```

pipeline {
    agent any

    parameters {
        // All parameters defined here
    }

    stages {
        stage('Display Parameters') {
            steps {
                script {
                    echo "STRING_PARAM: ${params.STRING_PARAM}"
                    echo "BOOLEAN_PARAM: ${params.BOOLEAN_PARAM}"
                    echo "CHOICE_PARAM: ${params.CHOICE_PARAM}"
                    echo "PASSWORD_PARAM: ${params.PASSWORD_PARAM}"
                    echo "FILE_PARAM: ${params.FILE_PARAM}"
                    echo "TEXT_PARAM: ${params.TEXT_PARAM}"
                    echo "CREDENTIALS_PARAM: ${params.CREDENTIALS_PARAM}"
                }
            }
        }
    }
}

```

- **Explanation:**
  - Show how to define and use parameters within pipeline stages.
  - Explain how to access and manipulate parameter values using the params object.

## 11. Advanced Parameter Handling

- **Parameter Validation:**

- Discuss how to add validation logic to ensure parameters are used correctly within the pipeline.
- **Dynamic Parameters:**
  - Explain techniques for generating or altering parameter values dynamically based on previous stages or conditions.

## 12. Practical Use Cases

- **Examples:**
  - Provide real-world examples where different parameter types are used together to create flexible and reusable pipelines.
  - Discuss scenarios like deploying to different environments, toggling features, or passing secure credentials for authentication.

## 13. Security Considerations

- **Best Practices:**
  - Emphasize the importance of securely handling sensitive parameters like passwords and credentials.
  - Recommend using Jenkins' built-in security features to manage and protect sensitive data.

## 14. Conclusion

- **Summary:**
  - Recap the key points about Jenkins pipeline parameters.
  - Encourage experimentation with different parameter types to enhance the flexibility of pipelines.